

**UNITED STATES DISTRICT COURT  
FOR THE SOUTHERN DISTRICT OF NEW YORK**

TOUCHTUNES MUSIC CORP.,

Plaintiff,

v.

ROWE INTERNATIONAL CORP.,  
ARACHNID, INC.,  
AMI ENTERTAINMENT, INC. and  
MERIT INDUSTRIES, INC. d/b/a/ MERIT  
ENTERTAINMENT,

Defendants.

Civil Action No. 07-cv-11450-RWS

AND RELATED COUNTERCLAIMS

**TOUCHTUNES' REPLY IN SUPPORT OF SUMMARY JUDGMENT OF  
NON-INFRINGEMENT OF THE '780 PATENT**

## TABLE OF CONTENTS

	<b><u>Page</u></b>
I. INTRODUCTION .....	1
II. ARGUMENT .....	2
A. The @data References Do Not Satisfy the “Non-Variable” Limitation of “Placeholder Tokenized Argument.” .....	2
B. The @data References Do Not Meet the “Single Customizable Argument Definition” Limitation of the Court’s Claim Construction.....	7
III. CONCLUSION.....	10

## TABLE OF AUTHORITIES

	<b><u>Page</u></b>
<i>Norian Corp. v. Stryker Corp.</i> , 432 F. 3d 1356, 1359 (Fed. Cir. 2005).....	9, 10

## I. INTRODUCTION

Arachnid's opposition to TouchTunes' motion for summary judgment of non-infringement of the '780 patent fails to identify any material facts in genuine dispute requiring trial. To the contrary, Arachnid makes key factual admissions that confirm non-infringement. Arachnid concedes that the alleged "placeholder tokenized arguments" (*i.e.*, "the @data references") are implemented in the accused TouchTunes system by a computer program written in the Ruby programming language, and that these @data references are *variables* in the Ruby programming language. (SMF Response, Dkt # 240 at ¶¶ 4, 7, 9 and 11). These admissions mandate summary judgment of non-infringement because the Court's claim construction of "placeholder tokenized argument" requires a "*non-variable* reference." (Markman, Dkt # 209 at 23-24). In fact, the Court held that Arachnid disclaimed the use of variable references to overcome the prior art. *Id.*

Arachnid also concedes that the accused TouchTunes system includes "*multiple possible*" argument definitions for the alleged placeholder tokenized argument. (Arachnid's Opposition ("Opp."), Dkt # 237 at 14). This admission also mandates summary judgment of non-infringement because the Court's claim construction requires that there be only a "*single*" argument definition for the placeholder tokenized argument. (Markman at 23-24). In fact, the Court held that Arachnid disclaimed systems with multiple possible argument definitions in order to overcome the prior art. *Id.* at 22-24.

Notwithstanding these fatal admissions, Arachnid attempts to maintain its baseless infringement claim by ignoring its prosecution disclaimers and the resulting claim construction of this Court. Specifically, Arachnid argues that the Court's

construction allows multiple possible argument definitions to be assigned to the placeholder as long as the placeholder's name does not change. (Opp. at 11). Of course, neither Arachnid's arguments to the PTO nor this Court's construction have anything to do with the "name" used for the placeholder. Arachnid further argues that each of the multiple possible argument definitions in TouchTunes' system constitutes a single argument definition and thereby satisfies the Court's construction. Of course, the clear purpose of the Court's construction was to exclude systems with multiple possible argument definitions.

Both of Arachnid's arguments fly in the face of the Court's claim construction and the express disclaimers Arachnid made to distinguish the prior art and obtain its patent. Arachnid's arguments are, at best, disingenuous and further highlight its willingness to bring and maintain frivolous infringement allegations against TouchTunes.

## II. ARGUMENT

### A. The @data References Do Not Satisfy the "Non-Variable" Limitation of "Placeholder Tokenized Argument."

Arachnid admits that the software program that implements the alleged placeholder tokenized arguments in TouchTunes' system is written in the Ruby programming language. (SMF Response at ¶ 4). Arachnid also admits that the particular and only references alleged by Arachnid to be placeholder tokenized arguments in TouchTunes' Ruby program are @data['message'], @data['item'] and @data. (SMF Response at ¶ 5). Arachnid further admits that each of these three @data references is a *variable* in the Ruby programming language. (SMF Response at ¶¶ 7, 9 and 11). On the basis of these admissions, TouchTunes is entitled to summary judgment of non-

infringement of the '780 patent because the Court's claim construction of "placeholder tokenized argument" (which applies to all claims of the '780 patent) requires these references to be "non-variable."

Despite these dispositive admissions, Arachnid argues that summary judgment should not be entered because, although the @data references are variables in the Ruby programming language, they are "non-variable" references as that term is used in the Court's construction. Arachnid attempts to convince this Court that the term "non-variable" in the Court's claim construction of placeholder tokenized argument "has nothing to do with computer programming variables." (Opp. at 1). Without citing any evidence for this theory, Arachnid asserts that the term "non-variable" as used by the Court in its claim construction "is not related to the term of art 'variable' used in computer programming." (Opp. at 11).

This assertion is directly contradicted by Arachnid's own argument to the PTO, where it distinguished the computer programming variable "n" from its claimed placeholder tokenized argument. (Schonfeld Dec., Dkt # 225, Ex. 7 at 12-13). Thus, the term "variable" has everything to do with computer programming variables, as it is the use of a computer programming variable that Arachnid disclaimed from its claim scope. Furthermore, the Court's claim construction requires that the multimedia segment structure containing the placeholder tokenized argument be "a portion of a program." (Markman at 12). The '780 patent is directed to computer programming, and Arachnid admits that the computer program that implements the alleged placeholder tokenized argument in the three accused display features is written in the Ruby computer programming language. (SMF Response at ¶4). Contrary to Arachnid's argument, the

term “non-variable” as used by the Court in its claim construction is necessarily tied to how the term “variable” is used in the accused Ruby computer programming language. Arachnid cannot simply ignore its disclaimer of computer programming variables or the computer language admittedly used by TouchTunes to implement the accused display features in order to avoid summary judgment of non-infringement.<sup>1</sup>

Arachnid further argues that, since the @data references have “unchanging reference names,” the references are not variables. (Opp. at 11). This argument is belied by the prosecution history and the Court’s construction. Whether or not something is a variable depends on whether its *value* changes, not whether its *name* changes. Arachnid made this perfectly clear during prosecution when it explained that Liu uses a variable named “n” that has a changing value. (Schonfeld Dec., Ex. 7 at 12-13). Under Arachnid’s new argument, the variable “n” is not a variable at all because its name is always “n” even though its value changes over time. This new argument obviously makes no sense and should be summarily rejected as an improper attempted to rewrite the prosecution history. In fact, the Court’s Markman Opinion specifically focuses on the *value* (not the name) of the placeholder tokenized argument when discussing Arachnid’s prosecution disclaimers:

TouchTunes correctly points out that Arachnid limited its claims in light of Liu such that the placeholder tokenized argument is not a variable, as in Liu, but instead a fixed value that can be replaced by a modifiable argument definition.” (Emphasis added).

---

<sup>1</sup> Arachnid argues that according to TouchTunes, since a Ruby variable can be defined as a constant by starting it with an uppercase letter (A-Z), constants are a type of variable in Ruby and, therefore, simply programming in the Ruby language precludes infringement. (Opp. at 12, n. 5). However, the opposite is true. By using a placeholder tokenized argument that starts with an uppercase letter, a non-variable reference to a single argument definition can be created in a Ruby program, but TouchTunes’ programs are simply not written that way, as Arachnid has admitted. (SMF Response at ¶¶ 4, 7, 9 and 11).

(Markman at 22).

It is beyond reasonable dispute that the critical issue here is whether the alleged placeholder tokenized arguments of the accused system are references to a single argument definition (making them non-variables) or to multiple argument definitions (making them variables). Arachnid admits, as it must, that all of the alleged placeholder tokenized arguments reference multiple argument definitions. (Opp. at 14). Dr. Schonfeld also confirms this fact. (*See, e.g.*, Schonfeld Dec. at ¶ 26 (“the value of the variable “@data[‘message’]” constantly changes in the Ruby program to reflect the current Bar Message that has been retrieved from the database”).<sup>2</sup>

In the Ruby programs that implement TouchTunes’ display features, the values assigned to the @data references are constantly changing over time to reflect the current definition that has been selected from a database. (Schonfeld Dec. at ¶ 26). Thus, at any given time, the @data references are assigned to one of numerous possible argument definitions. This is precisely the type of system disclosed by Liu and disclaimed by Arachnid in order to obtain its patent. Both Liu’s and TouchTunes’ systems are very different from the claimed invention, which uses a fixed placeholder tokenized argument (^TEST^) that is replaced by one, and only one, customizable argument definition (“This is a test message”). (Schonfeld Dec., Ex. 2, Fig. 2).

Nonetheless, Arachnid attempts to equate the replacement of ^TEST^ with “This is a test message” in the ’780 patent with the replacement of @data[‘message’] with “Tuesday is Ladies’ Night, Ladies Drink Free!” in TouchTunes’ Bar Messaging display feature. The comparison does not support Arachnid’s argument because the two systems

---

<sup>2</sup> Dr. Schonfeld provides this opinion in the context of the Bar Messaging display feature. (Schonfeld Dec. at ¶ 26). His opinion applies equally to the RSS Feed and Music Promotion display features. (Schonfeld Dec. at ¶¶ 36-37, 41-42).

are fundamentally different. In Arachnid's claimed system, ^TEST^ can only be replaced with the customizable argument definition "This is a test message." The claimed invention does not choose from among various other customizable argument definitions in the multimedia segment structure. In fact, Arachnid expressly disclaimed systems with multiple argument definitions in order to overcome the Examiner's prior art rejection over Liu. This Court made this clear in its Markman Opinion:

Arachnid also amended the claims to delete the language calling for "at least one argument definition corresponding to the tokenized argument." In its place, Arachnid added the narrower requirement for "*an* argument definition of the tokenized argument." Arachnid also added the limitation of "*the* argument definition customizing the predefined advertisement for local display." Thus, the claims no longer cover "at least one" definition "corresponding to" the tokenized argument. These amendments to the argument definition align with Arachnid's distinction of Liu and the amendment narrowing the tokenized argument to a "placeholder." (Italics supplied, citations omitted).

(Markman at 22-23).

Unlike the non-variable placeholder tokenized argument of Arachnid's claimed system, the values of TouchTunes' @data references constantly vary depending on what definition is retrieved from the database of multiple definitions. TouchTunes' system uses an incremental counter, like the counter used by Liu, to determine which of the multiple available definitions to select at any given time. (Schonfeld Dec. at ¶ 26). Thus, Arachnid's comparison is simply wrong. It is not the case that "@data ['message']" will necessarily display "Tuesday is Ladies' Night, Ladies Drink Free!" To the contrary, TouchTunes' system will display any one of the 10 available Bar Banners depending on the current value of the count variable. Arachnid's attempt to equate the claimed system with TouchTunes' system fails precisely because the @data references are variable tokenized arguments (like in Liu), while ^TEST^ is a fixed or non-variable "placeholder"



tokenized argument that can only reference a single customizable argument definition (*i.e.*, “This is a test message”). Accordingly, the @data references do not satisfy the “non-variable” limitation of the Court’s claim construction, thus entitling TouchTunes to summary judgment of non-infringement of the ’780 patent.

**B. The @data References Do Not Meet the “Single Customizable Argument Definition” Limitation of the Court’s Claim Construction.**

Arachnid admits that TouchTunes’ system chooses from among “*multiple* possible” argument definitions. (Opp. at 14). This admission alone is fatal to Arachnid’s infringement case. The Court’s claim construction makes clear that the claimed invention “no longer cover[s]” systems with more than one argument definition and, instead, is limited to a “*single* customizable argument definition.” (Markman at 22-24) (emphasis added). Arachnid argues that the accused system nonetheless meets the “single customizable argument definition” limitation because during operation, only one argument definition at a time is selected from the multiple possible definitions to replace the @data reference for display. Arachnid’s argument fails, as the clear purpose of the Court’s claim construction was to exclude systems with multiple possible argument definitions. *Id.*

Arachnid’s own “garage” and “car” analogy demonstrates the fallacy of its argument. This analogy considers a “garage” as being the placeholder tokenized argument and “cars” as being argument definitions. Because of Arachnid’s narrowing prosecution disclaimers, Arachnid’s garage is a special “placeholder” garage in which only one specific car may park. This car may be customized, *e.g.*, it may be painted a different color, the wheels may be changed, the fenders flared, but Arachnid’s placeholder garage is restricted to housing only this single particular car.

TouchTunes' system features a very different garage that can accommodate a variety of different cars over time. In the case of TouchTunes' Bar Messaging feature, one of 10 different cars may park in the garage at any given time. A counter, like the one in Liu, determines which car will be parked in the TouchTunes garage at any given time. In other words, unlike Arachnid's placeholder garage, TouchTunes' garage does not hold a place for a single customizable car. Instead, an entirely different car gets parked in TouchTunes' garage each time the counter is incremented. Thus, the TouchTunes variable garage is fundamentally different from Arachnid's non-variable "placeholder" garage which is limited to housing only a single customizable car.

Arachnid erroneously asserts that, according to TouchTunes' argument, putting a different car in the Arachnid garage would cause the garage to be "no longer a garage." (Opp. at 14). More accurately, however, it would mean that Arachnid's garage is no longer a "placeholder" garage for a single customizable car, as required by the Court's claim construction. Arachnid gave up a variable garage in order to get its patent and cannot now recapture such a variable garage in this litigation, as it is clearly attempting to do.

Arachnid argues for an interpretation of the Court's claim construction that would allow for the argument definition of the placeholder tokenized argument to be "one of many possible different definitions." (Opp. at 14). However, under the Court's claim construction, the placeholder tokenized argument may reference only a "single customizable argument definition." The Court's construction excludes systems in which the placeholder tokenized argument references one of many possible different definitions,

as that interpretation would improperly ignore Arachnid's prosecution disclaimers distinguishing Liu.

Recognizing that the Court's construction is fatal to its case, Arachnid suggests that the Court has erred in its construction and has failed to properly account for Federal Circuit law involving the use of the indefinite articles "a" and "an." (Opp. at 10, n.3). In particular, Arachnid erroneously asserts that the amendments and arguments Arachnid made to the PTO to overcome Liu were merely a "grammatical change" and should not limit the claims. To the contrary, as the Court correctly found, Arachnid's addition of a "placeholder" limitation for "*the* argument definition," and the replacement of the "*at least one* argument definition corresponding to the tokenized argument" with the narrower requirement for "*an* argument definition of the tokenized argument," were "significant" amendments and arguments which surrendered claim scope in order to overcome Liu. (Markman at 22-23). These amendments and associated arguments make clear that Arachnid intended to limit the invention to a single argument definition. The amendments were far more than a mere grammatical change, as Arachnid would now lead this Court to believe.

Thus, Arachnid has no basis for its belated attempt to circumvent its disclaimers or this Court's resulting construction. *See, Norian Corp. v. Stryker Corp.*, 432 F. 3d 1356, 1359 (Fed. Cir. 2005) (general rule that "a" or "an" means "one or more" in open-ended "comprising" claims does not apply when "the prosecution history shows that the term was used in its singular sense"). Although Arachnid seems to be suggesting in its argument that it did not need to make such broad disclaimers to overcome Liu (Opp. at 12), the fact is that it did make these broad disclaimers. As a result, the full scope of the

disclaimer must be taken into account when determining infringement, as this Court has properly done through its claim construction. *Id.* at 1361-62.

### III. CONCLUSION

It is beyond genuine dispute that the accused TouchTunes system does not meet the “non-variable reference” or the “single customizable argument definition” limitations of the Court’s construction of the claim term “placeholder tokenized argument.” Since these limitations apply to each and every claim of the ’780 patent, TouchTunes is entitled to summary judgment of non-infringement of the ’780 patent as a matter of law.

Respectfully submitted,

Date: December 10, 2012

By:           /James S. Blank/            
James S. Blank (JB-4376)  
**KAYE SCHOLER LLP**  
425 Park Avenue  
New York, New York 10022-3589  
Tel: (212) 836-8000  
Fax: (212) 836-8689  
E-mail: jblank@kayescholer.com

Joseph S. Presta  
E-mail: jsp@nixonvan.com  
Jonathon T. Reavill  
E-mail: jtr@nixonvan.com  
**NIXON & VANDERHYE, P.C.**  
901 North Glebe Road, 11<sup>th</sup> Floor  
Arlington, Virginia 22203  
Tel: (703) 816-4000  
Fax: (703) 816-4100

***Attorneys for Plaintiff***  
TouchTunes Music Corp.